L 11

```
PPPPPPPPPPPP        AAAAAAAAA        SSSSSSSSSSSS     CCCCCCCCCCCC     AAAAAAAAA     LLL
PPPPPPPPPPPP        AAAAAAAAA        SSSSSSSSSSSS     CCCCCCCCCCCC     AAAAAAAAA     LLL
PPPPPPPPPPPP        AAAAAAAAA        SSSSSSSSSSSS     CCCCCCCCCCCC     AAAAAAAAA     LLL
PPP         PPP  AAA         AAA  SSS              CCC              AAA         AAA  LLL
PPP         PPP  AAA         AAA  SSS              CCC              AAA         AAA  LLL
PPP         PPP  AAA         AAA  SSS              CCC              AAA         AAA  LLL
PPP         PPP  AAA         AAA  SSS              CCC              AAA         AAA  LLL
PPP         PPP  AAA         AAA  SSS              CCC              AAA         AAA  LLL
PPP         PPP  AAA         AAA  SSS              CCC              AAA         AAA  LLL
PPPPPPPPPPPP     AAA         AAA     SSSSSSSSS     CCC              AAA         AAA  LLL
PPPPPPPPPPPP     AAA         AAA     SSSSSSSSS     CCC              AAA         AAA  LLL
PPPPPPPPPPPP     AAA         AAA     SSSSSSSSS     CCC              AAA         AAA  LLL
PPP              AAAAAAAAAAAAAAA              SSS  CCC              AAAAAAAAAAAAAAA  LLL
PPP              AAAAAAAAAAAAAAA              SSS  CCC              AAAAAAAAAAAAAAA  LLL
PPP              AAAAAAAAAAAAAAA              SSS  CCC              AAAAAAAAAAAAAAA  LLL
PPP              AAA         AAA              SSS  CCC              AAA         AAA  LLL
PPP              AAA         AAA              SSS  CCC              AAA         AAA  LLL
PPP              AAA         AAA              SSS  CCC              AAA         AAA  LLL
PPP              AAA         AAA  SSSSSSSSSSSS        CCCCCCCCCCCC  AAA         AAA  LLLLLLLLLLLLLLL
PPP              AAA         AAA  SSSSSSSSSSSS        CCCCCCCCCCCC  AAA         AAA  LLLLLLLLLLLLLLL
PPP              AAA         AAA  SSSSSSSSSSSS        CCCCCCCCCCCC  AAA         AAA  LLLLLLLLLLLLLLL
```

FILEID**PASIO3

```
PPPPPPPP      AAAAAA      SSSSSSSS    IIIIII      000000      333333
PPPPPPPP      AAAAAA      SSSSSSSS    IIIIII      000000      333333
PP      PP  AA      AA  SS              II      00      00  33      33
PP      PP  AA      AA  SS              II      00      00  33      33
PP      PP  AA      AA  SS              II      00      00          33
PP      PP  AA      AA  SS              II      00      00          33
PPPPPPPP    AA      AA    SSSSSS        II      00      00        33
PPPPPPPP    AA      AA    SSSSSS        II      00      00        33
PP          AAAAAAAAAA          SS      II      00      00          33
PP          AAAAAAAAAA          SS      II      00      00          33
PP          AA      AA          SS      II      00      00  33      33      ....
PP          AA      AA          SS      II      00      00  33      33      ....
PP          AA      AA  SSSSSSSS    IIIIII      000000      333333        ....
PP          AA      AA  SSSSSSSS    IIIIII      000000      333333        ....

LL            IIIIII      SSSSSSSS
LL            IIIIII      SSSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II        SSSSSS
LL              II        SSSSSS
LL              II              SS
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

```
0000  ;
0000  ;****************************************************************************
0000  ;*                                                                          *
0000  ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
0000  ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
0000  ;*  ALL RIGHTS RESERVED.                                                    *
0000  ;*                                                                          *
0000  ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
0000  ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
0000  ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
0000  ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
0000  ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
0000  ;*  TRANSFERRED.                                                            *
0000  ;*                                                                          *
0000  ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
0000  ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
0000  ;*  CORPORATION.                                                            *
0000  ;*                                                                          *
0000  ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
0000  ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
0000  ;*                                                                          *
0000  ;*                                                                          *
0000  ;****************************************************************************
0000  ;
0000          .TITLE  PAS$IO_OUTPUT                   ; PASCAL RMS linkage
0000          .ident  'V04-000'
0000  ;
0000  ;**********************************************************************
0000  ;**********************************************************************
0000  ;**                                                                  **
0000  ;**                PASCAL RMS LINKAGE FOR VAX-11/780                 **
0000  ;**                =================================                 **
0000  ;**                                                                  **
0000  ;**                                                                  **
0000  ;**                  VERSION V1.2 -- JANUARY 1981                    **
0000  ;**                                                                  **
0000  ;**    DEVELOPED BY:  COMPUTER SCIENCE DEPARTMENT                     **
0000  ;**                   UNIVERSITY OF WASHINGTON                        **
0000  ;**                   SEATTLE, WA 98195                               **
0000  ;**                                                                  **
0000  ;**    AUTHORS:       MARK BAILEY, JOHN CHAN, HELLMUT GOLDE           **
0000  ;**                                                                  **
0000  ;**********************************************************************
0000  ;**********************************************************************
0000  ;
0000  ; Modified 08Jan80: 1) Allow output of 31 character scalar values in
0000  ;                   PAS$WRITESCAL.
0000  ;                   2) Bugfix in PAS$PUTBIN. Compiler was calling PAS$WRITEOK
0000  ;                   twice under some circumstances.
0000  ;                                  Paul Hohensee
0000  ;          05May80: Fix PAS$WRITESCAL to force output even if specified
0000  ;                   field width <= 0.
0000  ;                                  Paul Hohensee
0000  ;
0000  ;          16Oct80: Change PRN_CRLF so that lines are printed:
0000  ;                   <LF> <text> <CR>
0000  ;                                  Susan Azibert
```

```
0000  ;
0000  ;         13Jan81: Change all tests of status returns from RMS to
0000  ;                  BLBC R0,label and BLBS R0,label from CMPL R0,#RMS$NORMAL;
0000  ;                  BNEQ label, etc.
0000  ;                                  Paul Hohensee
0000  ;
0000  ;         13Jan81: Change scalar output so element name is right truncated
0000  ;                  for short field width, right justified and blank filled
0000  ;                  for long field width.
0000  ;
0000  ;         28Aug81: Use General addressing mode.  Steve Lionel.
0000  ;
0000  ;*************************************************************************
0000  ;*************************************************************************
0000  ;**                                                                     **
0000  ;**                                                                     **
0000  ;**                              SECTION 3                              **
0000  ;**                                                                     **
0000  ;**                         OUTPUT PROCEDURES                           **
0000  ;**                                                                     **
0000  ;**                                                                     **
0000  ;*************************************************************************
0000  ;*************************************************************************
0000  ;
0000  ;
0000  ; For any file variable the following storage is assumed:
0000  ;
0000  ;                      -------------------
0000  ;           FSB:      !      POINTER      !
0000  ;                      -------------------
0000  ;                     !    STATUS WORD    !
0000  ;                      -------------------
0000  ;                     !       LAST        !
0000  ;                      -------------------
0000  ;                     !     LINELIMIT     !
0000  ;                      -------------------
0000  ;                     !     LINECOUNT     !
0000  ;                      -------------------
0000  ;                     !   RECORD NUMBER   !
0000  ;                      -------------------
0000  ;          RAB:       !                   !
0000  ;                       44(HEX) BYTES
0000  ;                     !        .          !
0000  ;                              .
0000  ;                     !        .          !
0000  ;                      -------------------
0000  ;          FAB:       !                   !
0000  ;                       50(HEX) BYTES
0000  ;                     !        .          !
0000  ;                              .
0000  ;                     !        .          !
0000  ;                      -------------------
0000  ;          NAM:       !                   !   NOTE: The NAM block is allocated
0000  ;                       38(HEX) BYTES          for the PASCAL logical files
0000  ;                     !        .          !   'INPUT' and 'OUTPUT' only.
0000  ;                                .
0000  ;                     !                   !
```

```
                 0000  ;                    ------------------
                 0000  ;
                 0000  ; Macro options
                 0000  ;
                 0000          .DSABL  GBL                             ; no undefined references
                 0000          .ENABL  FPT                             ; rounded arithmetic
                 0000  ;
                 0000  ; External references
                 0000  ;
                 0000          .EXTRN  PAS$IOERROR
                 0000          .EXTRN  PAS$WRITEOK
                 0000          .EXTRN  PAS$BUFFEROVER
                 0000          .EXTRN  PAS$WRITELN
                 0000  ;
                 0000
                 0000          .EXTRN  FOR$CNV_OUT_D
                 0000          .EXTRN  FOR$CNV_OUT_E
                 0000          .EXTRN  FOR$CNV_OUT_F
                 0000          .EXTRN  FOR$CNV_OUT_I
                 0000          .EXTRN  FOR$CNV_OUT_O
                 0000          .EXTRN  FOR$CNV_OUT_Z
                 0000  ;
                 0000  ; Provide definitions of system values
                 0000  ;
                 0000          $DSCDEF                                 ; string descriptor definitions
                 0000          $FABDEF
                 0000          $RABDEF
                 0000          $RMSDEF                                 ; for status code checking
                 0000  ;
                 0000  ; PASCAL compiler constants
                 0000  ;
                 0000  ; Note: The constants below with the names 'PAS$C_XXXXX' are
                 0000  ;       used in the PASCAL compiler with the names 'XXXXX'. If the
                 0000  ;       values in the compiler are altered then the values below
                 0000  ;       must be altered accordingly.
                 0000  ;
                 0000  ;       PAS$C_DFLTRECSI = 257;                  ; default buffer size
                 0000  ;       PAS$C_NIL = 0                           ; NIL pointer
                 0000  ;       PAS$C_TRUE = 1                          ; TRUE
                 0000  ;       PAS$C_FALSE = 0                         ; FALSE
                 0000  ;       PAS$C_NOCARR = 0                        ; no carriage control
                 0000  ;       PAS$C_CARRIAGE = 1                      ; FORTRAN carriage control
                 0000  ;       PAS$C_LIST = 2                          ; LIST carriage control
                 0000  ;       PAS$C_PRN = 3                           ; PRN carriage control
                 0000  ;
                 0000  ; PRN carriage control constants
                 0000  ;
                 0000  ;       PRN_CRLF = ^X8D01                       ; PRN carriage control constant
                 0000                                                  ; for <LF> <text> <CR>
                 0000  ;       PRN_NULL = ^X0000                       ; PRN carriage control constant
                 0000                                                  ;     for no carriage control
                 0000  ;
                 0000  ; File status block constants
                 0000  ;
        00000018 0000          FSB$C_BLN = ^X18                        ; FSB block length
                 0000  ;       FSB$V_OPEN = 5
                 0000  ;       FSB$V_EOF = 1
```

```
                0000  ;        FSB$V_EOLN = 2
                0000  ;        FSB$V_GET = 3
                0000  ;        FSB$V_TXT = 4                       ; textfile flag
                0000  ;        FSB$V_RDLN = 0                      ; last access READLN
                0000  ;        FSB$V_DIR = 6                       ; direct access flag
                0000  ;        FSB$V_PUT = 7
                0000  ;        FSB$V_INT = 8                       ; internal flag
                0000  ;        FSB$V_PRMT = 9                      ; prompt flag
                0000  ;        FSB$V_OUTPUT = 10                   ; OUTPUT file flag
                0000  ;        FSB$V_ACTIN = 11                    ; actual input flag
                0000  ;        FSB$V_DELZ = 30                     ; delete file if empty
                0000  ;        FSB$V_INC = 31                      ; included file flag
                0000  ;        FSB$B_CC = 6                        ; carriage control byte offset
                0000  ;        FSB$M_OPEN = ^X0020
                0000  ;        FSB$M_EOF = ^X0002
                0000  ;        FSB$M_EOLN = ^X0004
                0000  ;        FSB$M_GET = ^X0008
                0000  ;        FSB$M_PRMT = ^X0200
                0000  ;        FSB$M_PUT = ^X00000080
                0000  ;        FSB$M_TXT = ^X0010
                0000  ;        FSB$M_RDLN = ^X0001
                0000  ;        FSB$M_DIR = ^X00000040
                0000  ;        FSB$M_INT = ^X00000100
                0000  ;        FSB$M_OUTPUT = ^X0400
                0000  ;        FSB$M_ACTIN = ^X0800
                0000  ;        FSB$M_DELZ = ^X40000000
                0000  ;        FSB$M_INC = ^X80000000
                0000  ;        FSB$L_CNT = 16                      ; line count (textfiles)
                0000  ;        FSB$L_INC = 20                      ; %INCLUDE block address
      0000000C  0000           FSB$L_LIM = 12                      ; linelimit
      00000008  0000           FSB$L_LST = 8                       ; last word offset
                0000  ;        FSB$L_PFSB = 20                     ; related file FSB for prompting
                0000                                               ; for INPUT, has address of OUTPUT FSB
                0000                                               ; for OUTPUT, has address of INPUT FSB
                0000                                               ; (shares storage with include address
                0000                                               ; and direct access record
                0000                                               ; buffer address)
                0000  ;        FSB$L_REC = 20                      ; record buffer address for
                0000                                               ; direct access (shares storage
                0000                                               ; with include address and related
                0000                                               ; file FSB)
                0000  ;        FSB$L_STA = 4                       ; status word offset
                0000  ;
                0000  ; Character constants
                0000  ;
                0000  ;        TAB = ^X09
      00000020  0000           SPACE = ^X20
                0000  ;        DOLLAR = ^X24
      0000000C  0000           FORMFEED = ^XC
      0000002A  0000           STAR = ^X2A
                0000  ;        PLUS = ^X2B
                0000  ;        MINUS = ^X2D
                0000  ;        POINT = ^X2E
      00000030  0000           ZERO = ^X30
      00000031  0000           ONE = ^X31
                0000  ;        NINE = ^X39
                0000  ;        AA = ^X41
```

```
                                 0000      ;          DD = ^X44
                                 0000      ;          EE = ^X45
                                 0000      ;          ZZ = ^X5A
                                 0000      ;          UNDERSCORE = ^X5F
                                 0000      ;          AA_SMALL = ^X61
                                 0000      ;          ZZ_SMALL = ^X7A
                                 0000      ;
                                 0000      ;
                         00000000          .PSECT  _PAS$CODE,              PIC,EXE,SHR,NOWRT
                                 0000      ;
                                 0000      ;          ********************
                                 0000      ;          *                  *
                                 0000      ;          *    PAS$PUTBIN    *
                                 0000      ;          *  pas$putbinary   *
                                 0000      ;          *                  *
                                 0000      ;          ********************
                                 0000      ;
                                 0000      ; Argument offsets
                                 0000      ;
                                 0000      ;          AP                          ; number of arguments (1)
                 00000004        0000               FSB_DISP = 04               ; FSB address
                                 0000      ;
                           00C0  0000               .ENTRY  PAS$PUTBIN,^M<R6,R7>
   00000000'GF      6C      FA   0002               CALLG   (AP),G^PAS$WRITEOK
                    02      11   0009               brb     newent
                           00C0  000B               .entry  pas$putbinary,^m<r6,r7>
                                 000D      newent:
        56    04 AC      D0      000D               MOVL    FSB_DISP(AP),R6             ; R6 = address of FSB
   57     18    56       C1      0011               ADDL3   R6,#FSB$C_BLN,R7            ; R7 = address of RAB
                                 0015               $PUT    RAB=R7
        04 A7    02      CA      001E               BICL2   #RAB$M_TPT,RAB$L_ROP(R7); clear TPT bit
             05 50       E9      0022               BLBC    R0,910$                     ; branch if error
        1E A7    00      90      0025               MOVB    #RAB$C_SEQ,RAB$B_RAC(R7); make sure sequential
                 04              0029               RET
                                 002A      ;
                                 002A      ; Write error
                                 002A      ;
                                 002A      910$:
                 50      DD      002A               PUSHL   R0
        7E    78 A7      9A      002C               MOVZBL  <RAB$C_BLN+FAB$B_FNS>(R7),-(SP)
              70 A7      DD      0030               PUSHL   <RAB$C_BLN+FAB$L_FNA>(R7)
   00000000'GF      03      FB   0033               CALLS   #3,G^PAS$IOERROR
                                 003A      ;
                                 003A      ;
                         0000003A          .PSECT  _PAS$CODE,              PIC,EXE,SHR,NOWRT
                                 003A      ;
                                 003A      ;          ********************
                                 003A      ;          *                  *
                                 003A      ;          *    PAS$PUTTXT    *
                                 003A      ;          *                  *
                                 003A      ;          ********************
                                 003A      ;
                                 003A      ; Increments the file pointer. If the pointer is positioned at the last
                                 003A      ; position at entry time then the buffer has overflowed.
                                 003A      ;
                                 003A      ; Argument offsets
                                 003A      ;
```

```
                              003A   ;           AP                            ; number of arguments (1)
                   00000004   003A               FSB_DISP = 04                 ; FSB address
                              003A   ;
                       000C   003A               .ENTRY PAS$PUTTXT,^M<R2,R3>
  00000000'GF    6C      FA   003C               CALLG    (AP),G^PAS$WRITEOK
         52    04 AC     D0   0043               MOVL     FSB_DISP(AP),R2      ; R2 = address of FSB
     53    18    52      C1   0047               ADDL3    R2,#FSB$C_BLN,R3     ; R3 = address of RAB
         08 A2   62      D1   004B               CMPL     (R2),FSB$L_LST(R2)
                 07      19   004F               BLSS     190$                 ; branch if ok
  00000000'GF    6C      FA   0051               CALLG    (AP),G^PAS$BUFFEROVER ; buffer overflow
                              0058         190$:
                 62      D6   0058               INCL     (R2)
                 04           005A               RET
                              005B   ;
                              005B   ;
                   0000005B                      .PSECT  _PAS$CODE,            PIC,EXE,SHR,NOWRT
                              005B   ;
                              005B   ;           ********************
                              005B   ;           *                  *
                              005B   ;           *  PAS$WRITECHAR   *
                              005B   ;           *                  *
                              005B   ;           ********************
                              005B   ;
                              005B   ; Writes a character to the file buffer. If the field width is less
                              005B   ; than or equal to zero then zero field width is used (ie. no output).
                              005B   ;
                              005B   ; Argument offsets
                              005B   ;
                              005B   ;           AP                            ; number of arguments (4)
                   00000004   005B               FSB_DISP = 04                 ; FSB address
                   00000008   005B               CHR_DISP = 08                 ; character value (low order byte)
                   0000000C   005B               FLD_DISP = 12                 ; field width (by value)
                   00000010   005B               NOT_DISP = 16                 ; (not used)
                              005B   ;
                       007C   005B               .ENTRY  PAS$WRITECHAR,^M<R2,R3,R4,R5,R6>
         56    04 AC     D0   005D               MOVL     FSB_DISP(AP),R6      ; R6 = address of FSB
                 56      DD   0061               PUSHL    R6
  00000000'GF    01      FB   0063               CALLS    #1,G^PAS$WRITEOK
              0C AC      D5   006A               TSTL     FLD_DISP(AP)         ; check field width
                 28      15   006D               BLEQ     199$                 ; exit if zero field width
                              006F   ;
                              006F   ; Check if enough room
                              006F   ;
     50    08 A6    66   C3   006F               SUBL3    (R6),FSB$L_LST(R6),R0 ; R0 = number of bytes left
         50    0C AC     D1   0074               CMPL     FLD_DISP(AP),R0
                 09      15   0078               BLEQ     110$
                 56      DD   007A               PUSHL    R6
  00000000'GF    01      FB   007C               CALLS    #1,G^PAS$BUFFEROVER  ; buffer overflow
                              0083         110$:
              0C AC      D7   0083               DECL     FLD_DISP(AP)
 0C AC   20    00 B6 00  2C   0086               MOVC5    #0,@(R6),#SPACE,FLD_DISP(AP),@(R6); blank fill
                 00 B6        008D
     63    08 AC    01   28   008F               MOVC3    #1,CHR_DISP(AP),(R3) ; put character
              66    53   D0   0094               MOVL     R3,(R6)              ; update file pointer
                              0097         199$:
                 04           0097               RET
                              0098   ;
```

```
                                      0098  ;
                                  00000098          .PSECT  _PAS$CODE,              PIC,EXE,SHR,NOWRT
                                      0098  ;
                                      0098  ;       *********************
                                      0098  ;       *                   *
                                      0098  ;       *   PAS$WRITESTR    *
                                      0098  ;       *                   *
                                      0098  ;       *********************
                                      0098  ;
                                      0098  ; Writes a string rigth justified with blank fill on the designated
                                      0098  ; file. If the field width is smaller than the string length the string
                                      0098  ; is truncated on the right.
                                      0098  ;
                                      0098  ; Argument offsets
                                      0098  ;
                                      0098  ;       AP                              ; number of arguments (4)
                            00000004  0098          FSB_DISP = 04                   ; FSB address
                            00000008  0098          STR_DISP = 08                   ; string address
                            0000000C  0098          FLD_DISP = 12                   ; field width (by value)
                            00000010  0098          LEN_DISP = 16                   ; string length (by value)
                                      0098  ;
                                00BC  0098          .ENTRY PAS$WRITESTR,^M<R2,R3,R4,R5,R7>
                   52    04 AC   D0   009A          MOVL    FSB_DISP(AP),R2         ; R2 = address of FSB
                            52   DD   009E          PUSHL   R2
            00000000'GF     01   FB   00A0          CALLS   #1,G^PAS$WRITEOK
                         0C AC   D5   00A7          TSTL    FLD_DISP(AP)
                            45   15   00AA          BLEQ    199$                    ; exit if field width <= 0
                                      00AC  ;
                                      00AC  ; Check if passing string value or address
                                      00AC  ;
                   04    10 AC   D1   00AC          CMPL    LEN_DISP(AP),#4
                            06   15   00B0          BLEQ    100$
                   57    08 AC   D0   00B2          MOVL    STR_DISP(AP),R7         ; R7 = address of string
                            04   11   00B6          BRB     101$
                                      00B8  100$:
                   57    08 AC   DE   00B8          MOVAL   STR_DISP(AP),R7         ; R7 = address of string
                                      00BC  101$:
              50    08 A2   62   C3   00BC          SUBL3   (R2),FSB$L_LST(R2),R0
                   50    0C AC   D1   00C1          CMPL    FLD_DISP(AP),R0
                            09   15   00C5          BLEQ    105$
                            52   DD   00C7          PUSHL   R2
            00000000'GF     01   FB   00C9          CALLS   #1,G^PAS$BUFFEROVER     ; buffer overflow
                                      00D0  105$:
         54    0C AC    10 AC   C3   00D0          SUBL3   LEN_DISP(AP),FLD_DISP(AP),R4; R4 = number of bytes to pad
                            08   1A   00D6          BGTRU   110$                    ; branch if padding required
         00 B2    67    0C AC   28   00D8          MOVC3   FLD_DISP(AP),(R7),@(R2) ; write width characters
                            0D   11   00DE          BRB     111$
                                      00E0  110$:                                   ; need to blank fill R4 bytes
   00 B2    54    20    00 B2   00   2C   00E0          MOVC5   #0,@(R2),#SPACE,R4,@(R2); blank fill
              63    67    10 AC   28   00E8          MOVC3   LEN_DISP(AP),(R7),(R3)  ; write string
                                      00ED  111$:                                   ; update pointers
                 04 BC    53   D0   00ED          MOVL    R3,@FSB_DISP(AP)
                                      00F1  199$:
                            04        00F1          RET
                                      00F2  ;
                                      00F2  ;
                                  000000F2          .PSECT  _PAS$CODE,              PIC,EXE,SHR,NOWRT
```

```
                             00F2   ;
                             00F2   ;       ********************
                             00F2   ;       *                  *
                             00F2   ;       *   PAS$WRITESCAL  *
                             00F2   ;       *                  *
                             00F2   ;       ********************
                             00F2   ;
                             00F2   ; Write out a scalar value on the designated text file. If the field
                             00F2   ; width is less than that required for the value, the value is left truncated
                             00F2   ; If the field width is greater than that required for the value, the
                             00F2   ; value is right justified with blank fill.
                             00F2   ;
                             00F2   ; Argument offsets
                             00F2   ;
                             00F2   ;       AP                              ; number of arguments (4)
                 00000004    00F2           FSB_DISP = 04                   ; FSB address
                 00000008    00F2           SCA_DISP = 08                   ; scalar value (by value)
                 0000000C    00F2           FLD_DISP = 12                   ; field width (by value)
                 00000010    00F2           NAM_DISP = 16                   ; namelist address
                 00000014    00F2           MAX_DISP = 20                   ; maximal ordinal value of
                             00F2                                           ; scalar (by value)
                             00F2   ;
                             00F2   ; Constants
                             00F2   ;
                 00000020    00F2           namelen = 32                    ; length in bytes of one entry in
                             00F2                                           ; name list.
                             00F2   ;
                     00FC    00F2           .ENTRY  PAS$WRITESCAL,^M<R2,R3,R4,R5,R6,R7>
        56    04 AC    D0    00F4           MOVL    FSB_DISP(AP),R6         ; R6 = address of FSB
              56       DD    00F8           PUSHL   R6
00000000'GF   01       FB    00FA           CALLS   #1,G^PAS$WRITEOK
   57   20    08 AC    C5    0101           MULL3   SCA_DISP(AP),#namelen,R7
        57    10 AC    C0    0106           ADDL2   NAM_DISP(AP),R7         ; R7 = scalar name address
                             010A   ;
                             010A   ; Calculate scalar name length and check for bounds
                             010A   ;
              08 AC    D5    010A           TSTL    SCA_DISP(AP)
              28       19    010D           BLSS    900$
     14 AC    08 AC    D1    010F           CMPL    SCA_DISP(AP),MAX_DISP(AP)
              21       14    0114           BGTR    900$
   67   20    20       3A    0116           LOCC    #SPACE,#namelen,(R7)
   51   20    50       C3    011A           SUBL3   R0,#namelen,R1
                             011E   ;
                             011E   ; Call PAS$WRITESTR to actually write the value to the buffer
                             011E   ;
              51       DD    011E           PUSHL   R1                      ; pass name length
              0C AC    DD    0120           PUSHL   FLD_DISP(AP)            ; pass field width
        04    51       D1    0123           CMPL    R1,#4                   ; pass by value or reference
              04       15    0126           BLEQ    110$
              57       DD    0128           PUSHL   R7                      ; by reference
              02       11    012A           BRB     111$
                             012C   110$:
              67       DD    012C           PUSHL   (R7)                    ; by value
                             012E   111$:
              04 AC    DD    012E           PUSHL   FSB_DISP(AP)
     FF62 CF  04       FB    0131           CALLS   #4,PAS$WRITESTR
                       04    0136           RET
```

```
                              0137  ;
                              0137  900$:
          7E  83A4 8F    3C   0137          MOVZWL  #^X83A4,-(SP)
          7E  0090 C6    9A   013C          MOVZBL  <FSB$C_BLN+RAB$C_BLN+FAB$B_FNS>(R6),-(SP)
              0088 C6    DD   0141          PUSHL   <FSB$C_BLN+RAB$C_BLN+FAB$L_FNA>(R6)
     00000000'GF    03   FB   0145          CALLS   #3,G^PAS$IOERROR
                              014C  ;
                              014C  ;
                      0000014C              .PSECT  _PAS$CODE,              PIC,EXE,SHR,NOWRT
                              014C  ;
                              014C  ;       ********************
                              014C  ;       *                  *
                              014C  ;       *   PAS$WRITEINT   *
                              014C  ;       *                  *
                              014C  ;       ********************
                              014C  ;
                              014C  ; Writes an integer right justified in the designated field width.
                              014C  ; If the field width is less than the minimum the minimum field width is
                              014C  ; used. If the integer won't fit in the designated field width then the
                              014C  ; amount needed is used. If the field width used will overflow the buffer
                              014C  ; a runtime error occurs.
                              014C  ;
                              014C  ; Argument offsets
                              014C  ;
                              014C          AP                                      ; number of arguments (4)
                    00000004  014C          FSB_DISP = 04                           ; FSB address
                    00000008  014C          INT_DISP = 08                           ; integer value
                    0000000C  014C          FLD_DISP = 12                           ; field width (by value)
                    00000010  014C          NOT_DISP = 16                           ; (not used)
                              014C  ;
                              014C  ; Other constants
                              014C  ;
                    00000001  014C          IMINP = 1                               ; minimum field width for
                              014C                                                  ; positive integers
                    00000002  014C          IMINN = 2                               ; minimum field width for
                              014C                                                  ; negative integers
                    00000014  014C          IMAX = 20                               ; maximum field width needed
                              014C                                                  ; for integers
                              014C  ;
                        03FC  014C          .ENTRY  PAS$WRITEINT,^M<R2,R3,R4,R5,R6,R7,R8,R9>
          56     04 AC   D0   014E          MOVL    FSB_DISP(AP),R6                 ; R6 = address of FSB
                    56   DD   0152          PUSHL   R6
     00000000'GF    01   FB   0154          CALLS   #1,G^PAS$WRITEOK
                              015B  ;
                              015B  ; Make room for descriptor on stack
                              015B  ;
          5E        08   C2   015B          SUBL2   #DSC$C_S_BLN,SP
          58        5E   D0   015E          MOVL    SP,R8                           ; R8 = address of descriptor
                              0161  ;
                              0161  ; Check for minimum field width (1 for positive, 2 for negative)
                              0161  ;
          53     0C AC   D0   0161          MOVL    FLD_DISP(AP),R3                 ; R3 = field width
                 08 AC   D5   0165          TSTL    INT_DISP(AP)                    ; test sign of value
                    0A   19   0168          BLSS    110$
                              016A                                                  ; positive value
          53        01   D1   016A          CMPL    #IMINP,R3                       ; use at least minimum
                    0D   15   016D          BLEQ    120$
```

```
PA
Sy
$$
$$
$1
$2
$2
$3
$3
$4
$4
$5
$6
CH
DI
DO
DS
DS
DS
EM
FA
FA
FA
FA
FL
FM
FM
FO
FO
FO
FO
FO
FO
FO
FS
FS
FS
FS
HM
IM
IM
IM
IN
LE
MA
NA
NA
NE
NO
OM
ON
OV
PA
PA
PA
PA
PA
PA
PA
```

```
              53   01   D0  016F          MOVL    #IMINP,R3
                   08   11  0172          BRB     120$
                            0174   110$:                               ; negative value
              53   02   D1  0174          CMPL    #IMINN,R3            ; use at least minimum
                   03   15  0177          BLEQ    120$
              53   02   D0  0179          MOVL    #IMINN,R3
                            017C   120$:                               ; R3 = field width
                            017C   ;
                            017C   ; Convert number to character string
                            017C   ;
     57   08 A6    66   C3  017C          SUBL3   (R6),FSB$L_LST(R6),R7   ; R7 = number of bytes left in line
              57   53   D1  0181          CMPL    R3,R7
                   09   15  0184          BLEQ    125$
                   56   DD  0186          PUSHL   R6
     00000000'GF   01   FB  0188          CALLS   #1,G^PAS$BUFFEROVER     ; buffer overflow
                            018F   125$:
              68   53   B0  018F          MOVW    R3,DSC$W_LENGTH(R8)     ; pass field width
           04 A8   66   D0  0192          MOVL    (R6),DSC$A_POINTER(R8)  ; pass buffer address
                   58   DD  0196          PUSHL   R8                      ; pass descriptor address
              08 AC     DD  0198          PUSHL   INT_DISP(AP)
     00000000'GF   02   FB  019B          CALLS   #2,G^FOR$CNV_OUT_I
              05   50   E9  01A2          BLBC    R0,130$
              66   53   C0  01A5          ADDL2   R3,(R6)                 ; update file pointer
                   36   11  01A8          BRB     199$                    ; exit, conversion succeeded
                            01AA   ;
                            01AA   ; Bad conversion; use a larger buffer and try again
                            01AA   ;
                            01AA   130$:
              68   14   B0  01AA          MOVW    #IMAX,DSC$W_LENGTH(R8)  ; pass buffer length
              5E   14   C2  01AD          SUBL2   #IMAX,SP                ; make room for buffer on stack
              59   5E   D0  01B0          MOVL    SP,R9
           04 A8   59   D0  01B3          MOVL    R9,DSC$A_POINTER(R8)    ; pass buffer address
                   58   DD  01B7          PUSHL   R8                      ; pass descriptor address
              08 AC     DD  01B9          PUSHL   INT_DISP(AP)
     00000000'GF   02   FB  01BC          CALLS   #2,G^FOR$CNV_OUT_I
              1B   50   E9  01C3          BLBC    R0,910$
        69   14    20   3B  01C6          SKPC    #SPACE,#IMAX,(R9)       ; skip leading spaces
                            01CA                                          ; R0 = number of remaining
                            01CA                                          ; characters
                            01CA                                          ; R1 = address of remaining
                            01CA                                          ; characters
              57   50   D1  01CA          CMPL    R0,R7                   ; check if enough room
                   09   15  01CD          BLEQ    140$
                   56   DD  01CF          PUSHL   R6
     00000000'GF   01   FB  01D1          CALLS   #1,G^PAS$BUFFEROVER     ; buffer overflow
                            01D8   140$:
     00 B6    61   50   28  01D8          MOVC3   R0,(R1),@(R6)           ; move string to output buffer
              66   53   D0  01DD          MOVL    R3,(R6)                 ; update file pointer
                            01E0   199$:
                        04  01E0          RET
                            01E1   ;
                            01E1   ; Output conversion error
                            01E1   ;
                            01E1   910$:
        7E   83A4  8F   3C  01E1          MOVZWL  #^X83A4,-(SP)
        7E   0090  C6   9A  01E6          MOVZBL  <FSB$C_BLN+RAB$C_BLN+FAB$B_FNS>(R6),-(SP)
             0088  C6   DD  01EB          PUSHL   <FSB$C_BLN+RAB$C_BLN+FAB$L_FNA>(R6)
```

```
PAS
Pse

PSE
---
;
$AB
_PA

Pha
---
Int
Com
Pas
Sym
Pas
Sym
Pse
Cro
Ass

The
499
The
115
14

Mac
---
_$2

772

The

MAC
```

```
     00000000'GF   03   FB  01EF          CALLS   #3,G^PAS$IOERROR
                            01F6    ;
                            01F6    ;
                        000001F6          .PSECT  _PAS$CODE,              PIC,EXE,SHR,NOWRT
                            01F6    ;
                            01F6    ;     *********************
                            01F6    ;     *                   *
                            01F6    ;     *  PAS$WRITEDOUBE   *
                            01F6    ;     *                   *
                            01F6    ;     *********************
                            01F6    ;
                            01F6    ; Write out a double precision number in 'E' format. A minimum
                            01F6    ; field width of FMIN is used.
                            01F6    ;
                            01F6    ; Argument offsets
                            01F6    ;
                            01F6          AP                              ; number of arguments (4)
                  00000004  01F6          FSB_DISP = 04                   ; FSB address
                  00000008  01F6          DOB_DISP = 08                   ; double number (by reference)
                  0000000C  01F6          FLD_DISP = 12                   ; field width (by value)
                            01F6    ;
                            01F6    ; Other constants
                            01F6    ;
                  00000008  01F6          FMIN = 8                        ; mimimum field width
                            01F6    ;
                      003C  01F6          .ENTRY  PAS$WRITEDOUBE,^M<R2,R3,R4,R5>
                            01F8    ;
                            01F8    ; Make room for descriptor and double precission value on stack
                            01F8    ;
               5E   10  C2  01F8          SUBL2   #<DSC$C_S_BLN+8>,SP
               51   5E  D0  01FB          MOVL    SP,R1                   ; R1 = descriptor address
         08 A1   08 BC  70  01FE          MOVD    @DOB_DISP(AP),DSC$C_S_BLN(R1); put value on stack
               55   01  D0  0203          MOVL    #1,R5                   ; set flag
                0010    31  0206          BRW     PAS$WREALE              ; jump to common code
                            0209    ;
                            0209    ;     *********************
                            0209    ;     *                   *
                            0209    ;     *  PAS$WRITEREALE   *
                            0209    ;     *                   *
                            0209    ;     *********************
                            0209    ;
                            0209    ; Write a real number in 'E' format. A minimum field width of EMIN is
                            0209    ; used.
                            0209    ;
                            0209    ; Argument offsets
                            0209    ;
                            0209          AP                              ; number of arguments (4)
                  00000004  0209          FSB_DISP = 04                   ; FSB address
                  00000008  0209          REL_DISP = 08                   ; real number (by value)
                  0000000C  0209          FLD_DISP = 12                   ; field width (by value)
                  00000010  0209          NOT_DISP = 16                   ; (not used)
                            0209    ;
                            0209    ; Other constants
                            0209    ;
                  00000008  0209          EMIN = 08                       ; minimum field width
                            0209    ;
                      003C  0209          .ENTRY  PAS$WRITEREALE,^M<R2,R3,R4,R5>
```

```
                        020B   ;
                        020B   ; Make room for descriptor and double precision value on stack
                        020B   ;
          5E    10  C2  020B          SUBL2   #<DSC$C_S_BLN + 8>,SP
          51    5E  D0  020E          MOVL    SP,R1
   08 A1     08 AC  56  0211          CVTFD   REL_DISP(AP),DSC$C_S_BLN(R1); put value on stack
          55    00  D0  0216          MOVL    #0,R5                     ; set flag
                        0219   ;
                        0219   ;>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
                        0219   ;
                        0219   ; The code below is common to both PAS$WRITEREALE and PAS$WRITEDOUBE.
                        0219   ; After the double precision value is placed on the stack there is no
                        0219   ; difference in the ways the values are converted.
                        0219   ;
                        0219   ;>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
                        0219   ;
                        0219    PAS$WREALE:
       52    04 AC  D0  0219          MOVL    FSB_DISP(AP),R2           ; R2 = address of FSB
                52  DD  021D          PUSHL   R2
  00000000'GF   01  FB  021F          CALLS   #1,G^PAS$WRITEOK
                        0226   ;
                        0226   ; Check field width and adjust if necessary
                        0226   ;
       53    0C AC  D0  0226          MOVL    FLD_DISP(AP),R3           ; R3 = field width
    54    53    07  C3  022A          SUBL3   #EMIN-1,R3,R4             ; R4 = number of digics to right
                06  14  022E          BGTR    110$                      ; branch if large enough field width
          53    08  D0  0230          MOVL    #EMIN,R3                  ; else set to minimum value
          54    01  D0  0233          MOVL    #1,R4                     ; and one digit to the right
                        0236   ;
                        0236   ; Check for buffer overflow and output the value
                        0236   ;
                        0236    110$:
  50    08 A2   62  C3  0236          SUBL3   (R2),FSB$L_LST(R2),R0
          50    53  D1  023B          CMPL    R3,R0
                09  15  023E          BLEQ    120$
                52  DD  0240          PUSHL   R2
  00000000'GF   01  FB  0242          CALLS   #1,G^PAS$BUFFEROVER       ; buffer overflow
                        0249    120$:
          61    53  B0  0249          MOVW    R3,DSC$W_LENGTH(R1)       ; store string length
       04 A1    62  D0  024C          MOVL    (R2),DSC$A_POINTER(R1)    ; store string address
                01  DD  0250          PUSHL   #1                        ; scale factor
                54  DD  0252          PUSHL   R4                        ; digits in fraction
                51  DD  0254          PUSHL   R1
             08 A1  DF  0256          PUSHAL  DSC$C_S_BLN(R1)           ; descriptor address
          55    00  D1  0259          CMPL    #0,R5                     ; check for single or double
                0E  12  025C          BNEQ    125$
  00000000'GF   04  FB  025E          CALLS   #4,G^FOR$CNV_OUT_E
                        0265   128$:
             0D 50  E9  0265          BLBC    R0,910$
          62    53  C0  0268          ADDL2   R3,(R2)
                    04  026B          RET
                        026C   ;
                        026C   125$:
  00C00000'GF   04  FB  026C          CALLS   #4,G^FOR$CNV_OUT_D
                F0  11  0273          BRB     128$
                        0275   ;
                        0275   ; Output conversion error
```

```
                                0275  ;
                                0275    910$:
          7E    83A4 8F    3C   0275        MOVZWL  #^X83A4,-(SP)
          7E    0090 C2    9A   027A        MOVZBL  <FSB$C_BLN+RAB$C_BLN+FAB$B_FNS>(R2),-(SP)
                0088 C2    DD   027F        PUSHL   <FSB$C_BLN+RAB$C_BLN+FAB$L_FNA>(R2)
     00000000'GF      03    FB   0283        CALLS   #3,G^PAS$IOERROR
                                028A  ;
                                028A  ;
                            0000028A        .PSECT  _PAS$CODE,                PIC,EXE,SHR,NOWRT
                                028A  ;
                                028A  ;     ********************
                                028A  ;     *                  *
                                028A  ;     *  PAS$WRITEDOUBF  *
                                028A  ;     *                  *
                                028A  ;     ********************
                                028A  ;
                                028A  ; Writes out a double number in fixed format.
                                028A  ;
                                028A  ; Argument offsets
                                028A  ;
                                028A        AP                                ; number of arguments (4)
                   00000004     028A        FSB_DISP = 04                     ; FSB address
                   00000008     028A        DOB_DISP = 08                     ; double value (by reference)
                   0000000C     028A        FLD_DISP = 12                     ; field width (by value)
                   00000010     028A        DIG_DISP = 16                     ; digits to right of decimal
                                028A                                          ; point (by value)
                                028A  ;
                                028A  ; Other constants
                                028A  ;
                   00000003     028A        FMIN = 3                          ; minimum field width
                   0000002A     028A        FMAX = 42                         ; maximum field width
                                028A  ;
                       007C     028A        .ENTRY  PAS$WRITEDOUBF,^M<R2,R3,R4,R5,R6>
                                028C  ;
                                028C  ; Make room for descriptor and double precision value on stack
                                028C  ;
           5E      10      C2   028C        SUBL2   #<DSC$C_S_BLN+8>,SP
           51      5E      D0   028F        MOVL    SP,R1                     ; R1 = address of descriptor
     08 A1      08 BC      70   0292        MOVD    @DOB_DISP(AP),DSC$C_S_BLN(R1); put value on stack
                  000D      31   0297        BRW     PAS$WREALF                ; jump to common code
                                029A  ;
                                029A  ;     ********************
                                029A  ;     *                  *
                                029A  ;     *  PAS$WRITEREALF  *
                                029A  ;     *                  *
                                029A  ;     ********************
                                029A  ;
                                029A  ; Writes out a real number in fixed format.
                                029A  ;
                                029A  ; Argument offsets
                                029A  ;
                                029A        AP                                ; number of arguments (4)
                   00000004     029A        FSB_DISP = 4                      ; FSB address
                   00000008     029A        REL_DISP = 08                     ; real number (by value)
                   0000000C     029A        FLD_DISP = 12                     ; field width
                   00000010     029A        DIG_DISP = 16                     ; digits to right of decimal
                                029A                                          ; point (by value)
```

```
                              029A   ;
                              029A   ; Other constants
                              029A   ;
                    00000003  029A          FMIN = 3                             ; minimum field width
                    0000002A  029A          FMAX = 42                            ; maximum field width
                              029A                                               ; (sign + point + 40)
                              029A   ;
                        007C  029A          .ENTRY  PAS$WRITEREALF,^M<R2,R3,R4,R5,R6>
                              029C   ;
                              029C   ; Make room for descriptor and double precision value on stack
                              029C   ;
             5E    10    C2   029C          SUBL2   #<DSC$C_S_BLN + 8>,SP
             51    5E    D0   029F          MOVL    SP,R1                        ; R1 = address of descriptor
      08 A1     08 AC    56   02A2          CVTFD   REL_DISP(AP),DSC$C_S_BLN(R1); store value on stack
                              02A7   ;
                              02A7   ;>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
                              02A7   ;
                              02A7   ; The code below is common to both PAS$WRITEREALF and PAS$WRITEDOUBF.
                              02A7   ; After the double precision value is placed on the stack there is no
                              02A7   ; difference in the way the values are converted.
                              02A7   ;
                              02A7   ;>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
                              02A7   ;
                              02A7   ; Check field widths and adjust if necessary
                              02A7   ;
                              02A7   PAS$WREALF:
         56     04 AC    D0   02A7          MOVL    FSB_DISP(AP),R6              ; R6 = address of FSB
                   56    DD   02AB          PUSHL   R6
    00000000'GF    01    FB   02AD          CALLS   #1,G^PAS$WRITEOK
         53     0C AC    D0   02B4          MOVL    FLD_DISP(AP),R3              ; R3 = field width (p)
                08 A1    73   02B8          TSTD    DSC$C_S_BLN(R1)              ; check if positive
                   0E    19   02BB          BLSS    105$
      00 B6        20    D0   02BD          MOVL    #SPACE,@(R6)                 ; if positive force blank
                04 AC    DD   02C1          PUSHL   FSB_DISP(AP)
    FD71 CF        01    FB   02C4          CALLS   #1,PAS$PUTTXT                ; write out blank
                   53    D7   02C9          DECL    R3                           ; adjust field width by 1
                              02CB   105$:
         54     10 AC    D0   02CB          MOVL    DIG_DISP(AP),R4              ; R4 = digits to right (q)
                   54    D5   02CF          TSTL    R4                           ; q > 0?
                   02    18   02D1          BGEQ    110$
                   54    D4   02D3          CLRL    R4                           ; q := 0
                              02D5   110$:
      50     53    54    C3   02D5          SUBL3   R4,R3,R0
             02    50    D1   02D9          CMPL    R0,#2                        ; p > q+2?
                   04    18   02DC          BGEQ    120$
      53     02    54    C1   02DE          ADDL3   R4,#2,R3                     ; p := q+2
                              02E2   ;
                              02E2   ; Set up descriptor and call conversion routine
                              02E2   ;
                              02E2   120$:
   55     08 A6    66    C3   02E2          SUBL3   (R6),FSB$L_LST(R6),R5        ; R5 = number of bytes left in line
             55    53    D1   02E7          CMPL    R3,R5
                   09    15   02EA          BLEQ    125$
                   56    DD   02EC          PUSHL   R6
    00000000'GF    01    FB   02EE          CALLS   #1,G^PAS$BUFFEROVER          ; buffer overflow
                              02F5   125$:
             61    53    B0   02F5          MOVW    R3,DSC$W_LENGTH(R1)
```

```
       04 A1    66   D0  02F8          MOVL    (R6),DSC$A_POINTER(R1)
                00   DD  02FC          PUSHL   #0                         ; scale factor
                54   DD  02FE          PUSHL   R4                         ; digits in fraction
                51   DD  0300          PUSHL   R1                         ; string descriptor
          08 A1      DF  0302          PUSHAL  DSC$C_S_BLN(R1)            ; value
 00000000'GF    04   FB  0305          CALLS   #4,G^FOR$CNV_OUT_F
          05 50      E9  030C          BLBC    R0,130$
          66    53   C0  030F          ADDL2   R3,(R6)                    ; update the file pointer
                3E   11  0312          BRB     199$
                         0314  ;
                         0314  ; Bad conversion; use a buffer of subfield+overflowsize and try again
                         0314  ;
                         0314  130$:
          51    5E   D0  0314          MOVL    SP,R1                      ; R1 = descriptor address
    53    54    2A   C1  0317          ADDL3   #FMAX,R4,R3                ; R3 = new buffer size
          5E    53   C2  031B          SUBL2   R3,SP                      ; make room on stack
          61    53   B0  031E          MOVW    R3,DSC$W_LENGTH(R1)
       04 A1    5E   D0  0321          MOVL    SP,DSC$A_POINTER(R1)       ; buffer address
                00   DD  0325          PUSHL   #0
                54   DD  0327          PUSHL   R4                         ; digits in fraction
                51   DD  0329          PUSHL   R1                         ; descriptor address
          08 A1      DF  032B          PUSHAL  DSC$C_S_BLN(R1)            ; value address
 00000000'GF    04   FB  032E          CALLS   #4,G^FOR$CNV_OUT_F
          1B 50      E9  0335          BLBC    R0,910$
    6E    53    20   3B  0338          SKPC    #SPACE,R3,(SP)             ; skip leading blanks
          55    50   D1  033C          CMPL    R0,R5
                09   15  033F          BLEQ    140$
                56   DD  0341          PUSHL   R6
 00000000'GF    01   FB  0343          CALLS   #1,G^PAS$BUFFEROVER        ; buffer overflow
                         034A  140$:
   00 B6  61    50   28  034A          MOVC3   R0,(R1),@(R6)              ; store string
          66    53   D0  034F          MOVL    R3,(R6)                    ; update file pointer
                         0352  199$:
                     04  0352          RET
                         0353  ;
                         0353  ; Output conversion error
                         0353  ;
                         0353  910$:
    7E    83A4 8F    3C  0353          MOVZWL  #^X83A4,-(SP)
    7E    0090 C6    9A  0358          MOVZBL  <FSB$C_BLN+RAB$C_BLN+FAB$B_FNS>(R6),-(SP)
          0088 C6    DD  035D          PUSHL   <FSB$C_BLN+RAB$C_BLN+FAB$L_FNA>(R6)
 00000000'GF    03   FB  0361          CALLS   #3,G^PAS$IOERROR
                     00000368          .PSECT  _PAS$CODE,                 PIC,EXE,SHR,NOWRT
                         0368  ;
                         0368  ;       *********************
                         0368  ;       *                   *
                         0368  ;       *   PAS$WRITEHEX    *
                         0368  ;       *                   *
                         0368  ;       *********************
                         0368  ;
                         0368  ; Writes out a longword in hexadecimal form. Leading zeros are printed
                         0368  ; up to eight places.
                         0368  ;
                         0368  ; Argument offsets
                         0368  ;
                         0368  ;      AP                                  ; number of arguments (4)
              00000004   0368          FSB_DISP = 04                      ; FSB address
```

```
                 00000008  0368         VAL_DISP = 08                        ; value to be printed
                 0000000C  0368         FLD_DISP = 12                        ; field width (by value)
                 00000010  0368         NOT_DISP = 16                        ; (not used)
                           0368   ;
                           0368   ; Other constants
                           0368   ;
                 00000008  0368         HMAX = 8                             ; maximum zero fill field width
                 00000010  0368         OVERFLOWSIZE = 16                    ; maximum overflow buffer size
                           0368   ;
                     00FC  0368         .ENTRY  PAS$WRITEHEX,^M<R2,R3,R4,R5,R6,R7>
           56   04 AC  D0  036A         MOVL    FSB_DISP(AP),R6              ; R6 = address of FSB
                   56  DD  036E         PUSHL   R6
      00000000'GF  01  FB  0370         CALLS   #1,G^PAS$WRITEOK
                0C AC  D5  0377         TSTL    FLD_DISP(AP)
                   65  15  037A         BLEQ    $40                          ; exit if field width <= 0
                           037C   ;
                           037C   ; Make room for descriptor on stack
                           037C   ;
              5E   08  C2  037C         SUBL2   #DSC$C_S_BLN,SP
              52   5E  D0  037F         MOVL    SP,R2                        ; R2 = address of descriptor
           62   0C AC  B0  0382         MOVW    FLD_DISP(AP),DSC$W_LENGTH(R2); store length
        04 A2      66  D0  0386         MOVL    (R6),DSC$A_POINTER(R2)       ; store buffer address
                   52  DD  038A         PUSHL   R2
                08 AC  DD  038C         PUSHL   VAL_DISP(AP)
      00000000'GF  02  FB  038F         CALLS   #2,G^FOR$CNV_OUT_Z
           66   0C AC  C0  0396         ADDL2   FLD_DISP(AP),(R6)
        04 B2      2A  91  039A         CMPB    #STAR,@DSC$A_POINTER(R2); test overflow
                   03  13  039E         BEQL    $25
                 0035  31  03A0         BRW     $35
                           03A3   $25:
           66   0C AC  C2  03A3         SUBL2   FLD_DISP(AP),(R6)            ; restore pointer
              62   10  B0  03A7         MOVW    #OVERFLOWSIZE,DSC$W_LENGTH(R2)
              5E   10  C2  03AA         SUBL2   #OVERFLOWSIZE,SP
              57   5E  D0  03AD         MOVL    SP,R7
        04 A2      57  D0  03B0         MOVL    R7,DSC$A_POINTER(R2)
                   52  DD  03B4         PUSHL   R2
                08 AC  DD  03B6         PUSHL   VAL_DISP(AP)
      00000000'GF  02  FB  03B9         CALLS   #2,G^FOR$CNV_OUT_Z
           67   10 20  3B  03C0         SKPC    #SPACE,#OVERFLOWSIZE,(R7); skip blanks
        54   50 0C AC  C3  03C4         SUBL3   FLD_DISP(AP),R0,R4
              51   54  C0  03C9         ADDL2   R4,R1
     00 B6   61 0C AC  28  03CC         MOVC3   FLD_DISP(AP),(R1),@(R6)      ; deposit string
           66   0C AC  C0  03D2         ADDL2   FLD_DISP(AP),(R6)            ; fix up pointer
                   09  11  03D6         BRB     $40
                           03D8   $35:
              54   08  CE  03D8         MNEGL   #HMAX,R4
      000004C6'EF      16  03DB         JSB     ZERO_FILL_R3
                       04  03E1   $40:  RET
                           03E2   ;
                           03E2   ;
                 000003E2  03E2         .PSECT  _PAS$CODE,                   PIC,EXE,SHR,NOWRT
                           03E2   ;
                           03E2   ;     ********************
                           03E2   ;     *                  *
                           03E2   ;     *  PAS$WRITEHEXD   *
                           03E2   ;     *                  *
                           03E2   ;     ********************
```

```
                          03E2  ;
                          03E2  ; Write out a double precision value (quadword) in hexadecimal form.
                          03E2  ; Leading zeros up to 16 places are printed
                          03E2  ;
                          03E2  ; Argument offsets
                          03E2  ;
                          03E2  ;      AP                                    ; number of arguments (4)
               00000004   03E2         FSB_DISP = 04                         ; FSB address
               00000008   03E2         VAR_DISP = 08                         ; value address
               0000000C   03E2         FLD_DISP = 12                         ; field width by value
               00000010   03E2         NOT_DISP = 16                         ; (not used)
                          03E2  ;
                          03E2  ; Other constants
                          03E2  ;
               00000008   03E2         HMAX = 8                              ; maximum field for leading zeros
                   0000   03E2         .ENTRY  PAS$WRITEHEXD,^M<>
   50    0C AC    08  C3  03E4         SUBL3   #HMAX,FLD_DISP(AP),R0         ; R0 = field width low bytes
                  06  14  03E9         BGTR    110$
         50    0C AC  D0  03EB         MOVL    FLD_DISP(AP),R0
                  16  11  03EF         BRB     111$
                          03F1  ;
                          03F1  ; Print low order longword
                          03F1  ;
                          03F1  110$:
                  00  DD  03F1         PUSHL   #0
                  50  DD  03F3         PUSHL   R0
   50    04    08 AC  C1  03F5         ADDL3   VAR_DISP(AP),#4,R0
                  60  DD  03FA         PUSHL   (R0)                          ; low order longword
               04 AC  DD  03FC         PUSHL   FSB_DISP(AP)
      FF64 CF     04  FB  03FF         CALLS   #4,PAS$WRITEHEX
           50     08  D0  0404         MOVL    #HMAX,R0                      ; field width high bytes
                          0407  ;
                          0407  ; Print R0 digits of high order longword
                          0407  ;
                          0407  111$:
                  00  DD  0407         PUSHL   #0
                  50  DD  0409         PUSHL   R0
               08 BC  DD  040B         PUSHL   @VAR_DISP(AP)
               04 AC  DD  040E         PUSHL   FSB_DISP(AP)
      FF52 CF     04  FB  0411         CALLS   #4,PAS$WRITEHEX
                      04  0416         RET
                          0417  ;
                          0417  ;
               00000417   0417         .PSECT  _PAS$CODE,                    PIC,EXE,SHR,NOWRT
                          0417  ;
                          0417  ;      ********************
                          0417  ;      *                  *
                          0417  ;      *   PAS$WRITEOCT   *
                          0417  ;      *                  *
                          0417  ;      ********************
                          0417  ;
                          0417  ; Argument offsets
                          0417  ;
                          0417  ;      AP                                    ; number of arguments (4)
               00000004   0417         FSB_DISP = 04                         ; FSB address
               00000008   0417         VAL_DISP = 08                         ; value to be printed
               0000000C   0417         FLD_DISP = 12                         ; field width
```

```
                00000010  0417          NOT_DISP = 16                     ; (not used)
                          0417  ;
                          0417  ; Other constants
                          0417  ;
                0000000B  0417          OMAX = 11                         ; maximum field for leading zeros
                00000014  0417          OVERFLOWSIZE = 20                 ; overflow buffer size
                          0417  ;
                    00FC  0417          .ENTRY  PAS$WRITEOCT,^M<R2,R3,R4,R5,R6,R7>
         56    04 AC   D0  0419          MOVL    FSB_DISP(AP),R6           ; R6 = address of FSB
                  56   DD  041D          PUSHL   R6
     00000000'GF   01   FB  041F          CALLS   #1,G^PAS$WRITEOK
            0C AC      D5  0426          TSTL    FLD_DISP(AP)
               65      15  0429          BLEQ    $43                       ; exit if field width <= 0
                          042B  ;
                          042B  ; Make room for descriptor on stack
                          042B  ;
         5E    08      C2  042B          SUBL2   #DSC$C_S_BLN,SP
         52    5E      D0  042E          MOVL    SP,R2
         62    0C AC   B0  0431          MOVW    FLD_DISP(AP),DSC$W_LENGTH(R2); store length
      04 A2    66      D0  0435          MOVL    (R6),DSC$A_POINTER(R2)  ; store buffer address
               52      DD  0439          PUSHL   R2
            08 AC      DD  043B          PUSHL   VAL_DISP(AP)
     00000000'GF   02   FB  043E          CALLS   #2,G^FOR$CNV_OUT_O
         66    0C AC   C0  0445          ADDL2   FLD_DISP(AP),(R6)
      04 B2    2A      91  0449          CMPB    #STAR,@DSC$A_POINTER(R2); test overflow
               03      13  044D          BEQL    $55
             0035      31  044F          BRW     $65
                          0452  $55:
         66    0C AC   C2  0452          SUBL2   FLD_DISP(AP),(R6)        ; restore pointer
         62    14      B0  0456          MOVW    #OVERFLOWSIZE,DSC$W_LENGTH(R2)
         5E    14      C2  0459          SUBL2   #OVERFLOWSIZE,SP
         57    5E      D0  045C          MOVL    SP,R7
      04 A2    57      D0  045F          MOVL    R7,DSC$A_POINTER(R2)
               52      DD  0463          PUSHL   R2
            08 AC      DD  0465          PUSHL   VAL_DISP(AP)
     00000000'GF   02   FB  0468          CALLS   #2,G^FOR$CNV_OUT_O
      67    14    20   3B  046F          SKPC    #SPACE,#OVERFLOWSIZE,(R7); skip blanks
   54    50    0C AC   C3  0473          SUBL3   FLD_DISP(AP),R0,R4
         51    54      C0  0478          ADDL2   R4,R1
00 B6    61    0C AC   28  047B          MOVC3   FLD_DISP(AP),(R1),@(R6) ; deposit string
         66    0C AC   C0  0481          ADDL2   FLD_DISP(AP),(R6)       ; fix up pointer
               09      11  0485          BRB     $43
                          0487  $65:
         54    0B      CE  0487          MNEGL   #OMAX,R4
       000004C6'EF     16  048A          JSB     ZERO_FILL_R3
                       04  0490  $43:    RET
                          0491  ;
                          0491  ;
                00000491  0491          .PSECT  _PAS$CODE,              PIC,EXE,SHR,NOWRT
                          0491  ;
                          0491  ;       *********************
                          0491  ;       *                   *
                          0491  ;       *  PAS$WRITEOCTD    *
                          0491  ;       *                   *
                          0491  ;       *********************
                          0491  ;
                          0491  ; Write out a double precision value (quadword) in octal format.
```

```
                           0491  ; Leading zeros up to twenty-two places are printed.
                           0491  ;
                           0491  ; Argument offsets
                           0491  ;
                           0491  ;         AP                                   ; number of arguments (4)
                 00000004  0491            FSB_DISP = 04                        ; FSB address
                 00000008  0491            VAR_DISP = 08                        ; value address
                 0000000C  0491            FLD_DISP = 12                        ; field width by value
                 00000010  0491            NOT_DISP = 16                        ; (not used)
                           0491  ;
                           0491  ; Other constants
                           0491  ;
                 0000000B  0491            OMAX = 11                            ; maximum field for leading zeros
                           0491  ;
                     0000  0491            .ENTRY  PAS$WRITEOCTD,^M<>
  50   0C AC    0B   C3    0493            SUBL3   #OMAX,FLD_DISP(AP),R0        ; R0 = field width low bytes
                06   14    0498            BGTR    110$
       50   0C AC   D0     049A            MOVL    FLD_DISP(AP),R0
                16   11    049E            BRB     111$
                           04A0  ;
                           04A0  ; Print low order longword
                           04A0  ;
                           04A0  110$:
                00   DD    04A0            PUSHL   #0
                50   DD    04A2            PUSHL   R0
  50   04    08 AC   C1    04A4            ADDL3   VAR_DISP(AP),#4,R0
                60   DD    04A9            PUSHL   (R0)                         ; low order longword
             04 AC   DD    04AB            PUSHL   FSB_DISP(AP)
    FF64 CF     04   FB    04AE            CALLS   #4,PAS$WRITEOCT
          50    0B   D0    04B3            MOVL    #OMAX,R0                     ; field width high bytes
                           04B6  ;
                           04B6  ; Print R0 digits of high order longword
                           04B6  ;
                           04B6  111$:
                00   DD    04B6            PUSHL   #0
                50   DD    04B8            PUSHL   R0
             08 BC   DD    04BA            PUSHL   @VAR_DISP(AP)
             04 AC   DD    04BD            PUSHL   FSB_DISP(AP)
    FF52 CF     04   FB    04C0            CALLS   #4,PAS$WRITEOCT
                     04    04C5            RET
                           04C6  ;
                           04C6  ;
                 000004C6  04C6            .PSECT  _PAS$CODE,           PIC,EXE,SHR,NOWRT
                           04C6  ;
                           04C6  ;         ********************
                           04C6  ;         *                  *
                           04C6  ;         *   ZERO_FILL_R3   *
                           04C6  ;         *                  *
                           04C6  ;         ********************
                           04C6  ;
                           04C6  ; JSB routine to zero-fill octal and hex output
                           04C6  ;
                           04C6  ;
                           04C6  ZERO_FILL_R3:                                ; entry point
  52   0C AC        CE     04C6            MNEGL   FLD_DISP(AP),R2              ; get length
       54   52      D1     04CA            CMPL    R2,R4
            03      18     04CD            BGEQ    $30
```

```
                52   54   D0  04CF              MOVL    R4,R2
         53   04 BC   D0  04D2     $30:    MOVL    @FSB_DISP(AP),R3        ; move address to R3
           6342   20   91  04D6     $10:    CMPB    #SPACE,(R3)[R2]         ; check next byte for blank
                  0C   12  04DA             BNEQ    $20                     ; done if not blank
           6342   30   90  04DC             MOVB    #ZERO,(R3)[R2]          ; put in zero
   EE 52   FFFFFFFF 8F   F2  04E0             AOBLSS  #-1,R2,$10
                  05       04E8     $20:    RSB                             ; return
                           04E9    ;
                           04E9    ;
              000004E9             .PSECT  _PAS$CODE,              PIC,EXE,SHR,NOWRT
                           04E9    ;
                           04E9    ;       *********************
                           04E9    ;       *                   *
                           04E9    ;       *  PAS$LINELIMIT    *
                           04E9    ;       *                   *
                           04E9    ;       *********************
                           04E9    ;
                           04E9    ; Sets the linelimit for a given file.
                           04E9    ;
                           04E9    ; Argument offsets
                           04E9    ;
                           04E9    ;       AP                              ; number of arguments (2)
             00000004      04E9            FSB_DISP = 04                   ; FSB address
             00000008      04E9            VAL_DISP = 08                   ; linelimit value
                           04E9    ;
                  0004     04E9            .ENTRY  PAS$LINELIMIT,^M<R2>
         52   08 AC   D0  04EB            MOVL    8(AP),R2
      0C A2   04 AC   D0  04EF            MOVL    4(AP),FSB$L_LIM(R2)
                  04       04F4            RET
                           04F5    ;
                           04F5    ;
              000004F5             .PSECT  _PAS$CODE,              PIC,EXE,SHR,NOWRT
                           04F5    ;
                           04F5    ;       *********************
                           04F5    ;       *                   *
                           04F5    ;       *     PAS$PAGE      *
                           04F5    ;       *                   *
                           04F5    ;       *********************
                           04F5    ;
                           04F5    ; Writes a page eject character (1H1 or FORMFEED) to the designated file.
                           04F5    ;
                           04F5    ; Arguments offsets
                           04F5    ;
                           04F5    ;       AP                              ; number of arguments (1)
             00000004      04F5            FSB_DISP = 04                   ; FSB address
                           04F5    ;
                  001C     04F5            .ENTRY  PAS$PAGE,^M<R2,R3,R4>
         52   04 AC   D0  04F7            MOVL    FSB_DISP(AP),R2         ; R2 = FSB address
   53   0000005C 8F   52   C1  04FB            ADDL3   R2,#<FSB$C_BLN+RAB$C_BLN>,R3; R3 = FAB address
         54   18   52   C1  0503            ADDL3   R2,#FSB$C_BLN,R4        ; R4 = RAB address
         62   28 A4   D1  0507            CMPL    RAB$L_RBF(R4),(R2)
                  09   13  050B            BEQL    10$
                  52   DD  050D            PUSHL   R2
        00000000'EF   01   FB  050F            CALLS   #1,PAS$WRITELN          ; terminate current line
                           0516     10$:
                  00   DD  0516            PUSHL   #0                      ; fill
                  01   DD  0518            PUSHL   #1                      ; field width
```

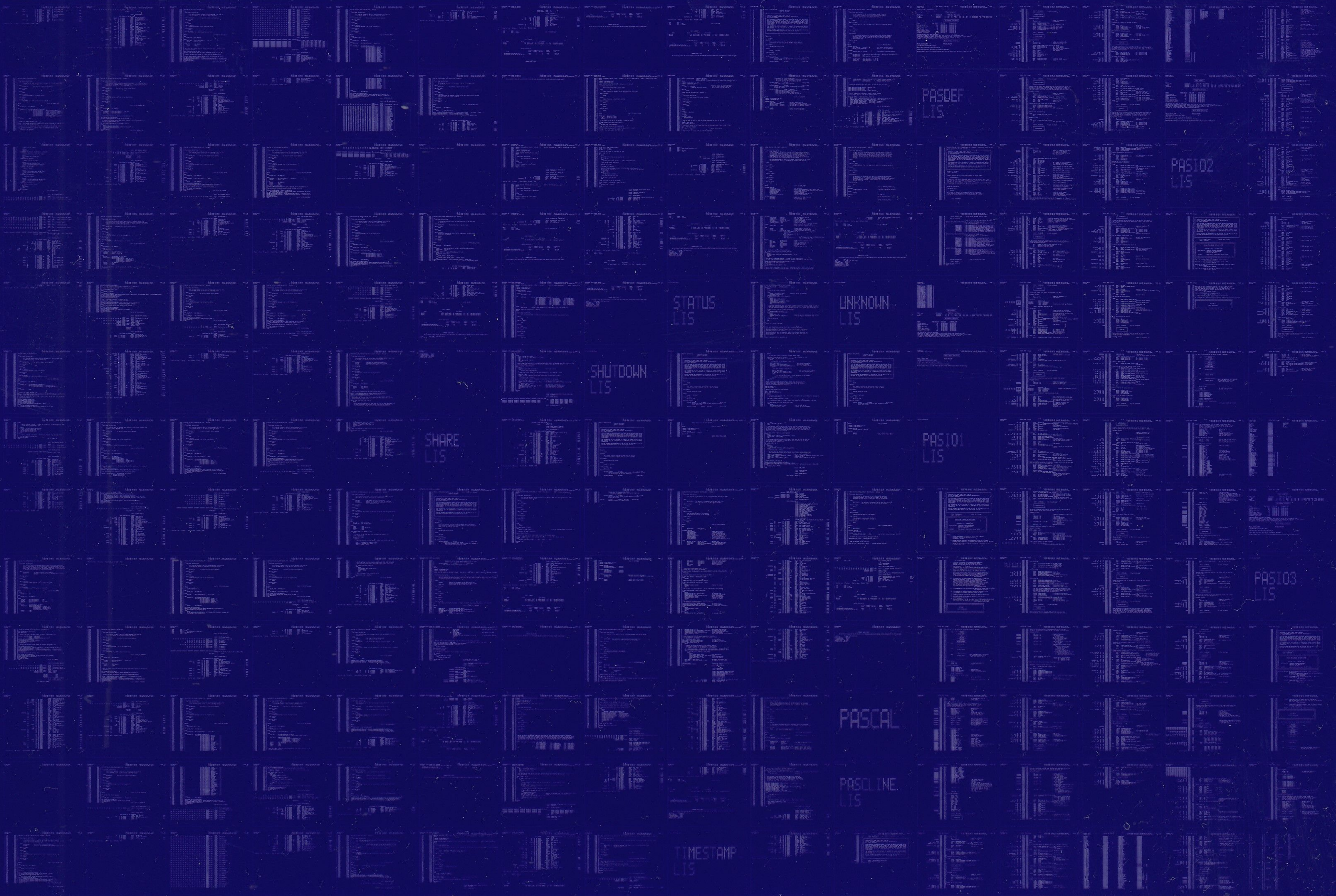
```
          04 1E A3   00   E0  051A        BBS     #FAB$V_FTN,FAB$B_RAT(R3),20$; check for carriage control
                     0C   DD  051F        PUSHL   #FORMFEED                   ; not FORTRAN
                     02   11  0521        BRB     30$
                              0523  20$:
                     31   DD  0523        PUSHL   #ONE                        ; FORTRAN
                              0525  30$:
                     52   DD  0525        PUSHL   R2                          ; FSB address
             FB2F CF 04   FB  0527        CALLS   #4,PAS$WRITECHAR
                     52   DD  052C        PUSHL   R2
         00000000'EF 01   FB  052E        CALLS   #1,PAS$WRITELN              ; terminate line
                          04  0535        RET                                 ; return
                              0536  ;
                              0536  ;
                              0536  ;
                              0536        .END
```

```
PAS$IO_OUTPUT                     ; PASCAL RMS linkage                      16-SEP-1984 02:07:46  VAX/VMS Macro V04-00        Page  22
Symbol table                                                                  5-SEP-1984 02:32:22  [PASCAL.SRC]PASIO3.MAR;1          (1)

$$.TMP1                         = 00000001                 PAS$WREALE                      00000219 R       02
$$.TMP2                         = 00000057                 PAS$WREALF                      000002A7 R       02
$10                               000004D6 R       02      PAS$WRITECHAR                   0000005B RG      02
$20                               000004E8 R       02      PAS$WRITEDOUBE                  000001F6 RG      02
$25                               000003A3 R       02      PAS$WRITEDOUBF                  0000028A RG      02
$30                               000004D2 R       02      PAS$WRITEHEX                    00000368 RG      02
$35                               000003D8 R       02      PAS$WRITEHEXD                   000003E2 RG      02
$40                               000003E1 R       02      PAS$WRITEINT                    0000014C RG      02
$43                               00000490 R       02      PAS$WRITELN                     ********    X    00
$55                               00000452 R       02      PAS$WRITEOCT                    00000417 RG      02
$65                               00000487 R       02      PAS$WRITEOCTD                   00000491 RG      02
CHR_DISP                        = 00000008                 PAS$WRITEOK                     ********    X    00
DIG_DISP                        = 00000010                 PAS$WRITEREALE                  00000209 RG      02
DOB_DISP                        = 00000008                 PAS$WRITEREALF                  0000029A RG      02
DSC$A_POINTER                   = 00000004                 PAS$WRITESCAL                   000000F2 RG      02
DSC$C_S_BLN                     = 00000008                 PAS$WRITESTR                    00000098 RG      02
DSC$W_LENGTH                    = 00000000                 RAB$B_RAC                     = 0000001E
EMIN                            = 00000008                 RAB$C_BLN                     = 00000044
FAB$B_FNS                       = 00000034                 RAB$C_SEQ                     = 00000000
FAB$B_RAT                       = 0000001E                 RAB$L_RBF                     = 00000028
FAB$L_FNA                       = 0000002C                 RAB$L_ROP                     = 00000004
FAB$V_FTN                       = 00000000                 RAB$M_TPT                     = 00000002
FLD_DISP                        = 0000000C                 REL_DISP                      = 00000008
FMAX                            = 0000002A                 SCA_DISP                      = 00000008
FMIN                            = 00000003                 SPACE                         = 00000020
FOR$CNV_OUT_D                     ********    X    00      STAR                          = 0000002A
FOR$CNV_OUT_E                     ********    X    00      STR_DISP                      = 00000008
FOR$CNV_OUT_F                     ********    X    00      SYS$PUT                         ********   G     02
FOR$CNV_OUT_I                     ********    X    00      VAL_DISP                      = 00000008
FOR$CNV_OUT_O                     ********    X    00      VAR_DISP                      = 00000008
FOR$CNV_OUT_Z                     ********    X    00      ZERO                          = 00000030
FORMFEED                        = 0000000C                 ZERO_FILL_R3                    000004C6 R       02
FSB$C_BLN                       = 00000018
FSB$L_LIM                       = 0000000C
FSB$L_LST                       = 00000008
FSB_DISP                        = 00000004
HMAX                            = 00000008
IMAX                            = 00000014
IMINN                           = 00000002
IMINP                           = 00000001
INT_DISP                        = 00000008
LEN_DISP                        = 00000010
MAX_DISP                        = 00000014
NAMELEN                         = 00000020
NAM_DISP                        = 00000010
NEWENT                            0000000D R       02
NOT_DISP                        = 00000010
OMAX                            = 0000000B
ONE                             = 00000031
OVERFLOWSIZE                    = 00000014
PAS$BUFFEROVER                    ********    X    00
PAS$IOERROR                       ********    X    00
PAS$LINELIMIT                     000004E9 RG      02
PAS$PAGE                          000004F5 RG      02
PAS$PUTBIN                        00000000 RG      02
PAS$PUTBINARY                     0000000B RG      02
PAS$PUTTXT                        0000003A RG      02
```

```
                                           +----------------+
                                           ! Psect synopsis !
                                           +----------------+

PSECT name                 Allocation          PSECT No.  Attributes
----------                 ----------          ---------  ----------
.  ABS  .                  00000000  (     0.)  00 (  0.)  NOPIC   USR   CON   ABS   LCL NOSHR NOEXE NORD   NOWRT NOVEC BYTE
$ABS$                      00000000  (     0.)  01 (  1.)  NOPIC   USR   CON   ABS   LCL NOSHR   EXE   RD     WRT NOVEC BYTE
_PAS$CODE                  00000536  (  1334.)  02 (  2.)    PIC   USR   CON   REL   LCL   SHR   EXE   RD   NOWRT NOVEC BYTE

                                        +------------------------+
                                        ! Performance indicators !
                                        +------------------------+

Phase                       Page faults    CPU Time       Elapsed Time
-----                       -----------    --------       ------------
Initialization                      34     00:00:00.08    00:00:00.50
Command processing                 135     00:00:00.45    00:00:01.24
Pass 1                             252     00:00:08.14    00:00:16.17
Symbol table sort                    0     00:00:00.87    00:00:00.91
Pass 2                             194     00:00:02.86    00:00:04.53
Symbol table output                 12     00:00:00.07    00:00:00.09
Psect synopsis output                3     00:00:00.03    00:00:00.02
Cross-reference output               0     00:00:00.00    00:00:00.00
Assembler run totals               633     00:00:12.50    00:00:23.47

The working set limit was 1500 pages.
49931 bytes (98 pages) of virtual memory were used to buffer the intermediate code.
There were 40 pages of symbol table space allocated to hold 726 non-local and 40 local symbols.
1154 source lines were read in Pass 1, producing 61 object records in Pass 2.
14 pages of virtual memory were used to define 12 macros.

                                       +--------------------------+
                                       ! Macro library statistics !
                                       +--------------------------+

Macro library name                     Macros defined
------------------                     --------------
_$255$DUA28:[SYSLIB]STARLET.MLB;2                 9

772 GETS were required to define 9 macros.

There were no errors, warnings or information messages.

MACRO/DISABLE=TRACE/LIS=LIS$:PASIO3/OBJ=OBJ$:PASIO3 MSRC$:PASIO3/UPDATE=(ENH$:PASIO3)
```

0292 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY



0293 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

PASRT1
LIS

PASMACROS
REQ

PASFCB
SDL

PASPROLOG
REQ

PASOPEDEF
REQ

PASABSL
LIS

PASCARD2
LIS

PASRTL

PASBUGCOD
REQ

PASCLOSE2
LIS

PASRT2
LIS

PASRT3
LIS

PASRTL
MAP

PASIO4
LIS

PASEXTERN
REQ

PASKDB
REQ

PASPFD
REQ

PASCONVER
LIS

PASLIB
REQ

PASPFU
REQ

PASBIN
LIS

PASCLOCK2
LIS

PASRT4
LIS